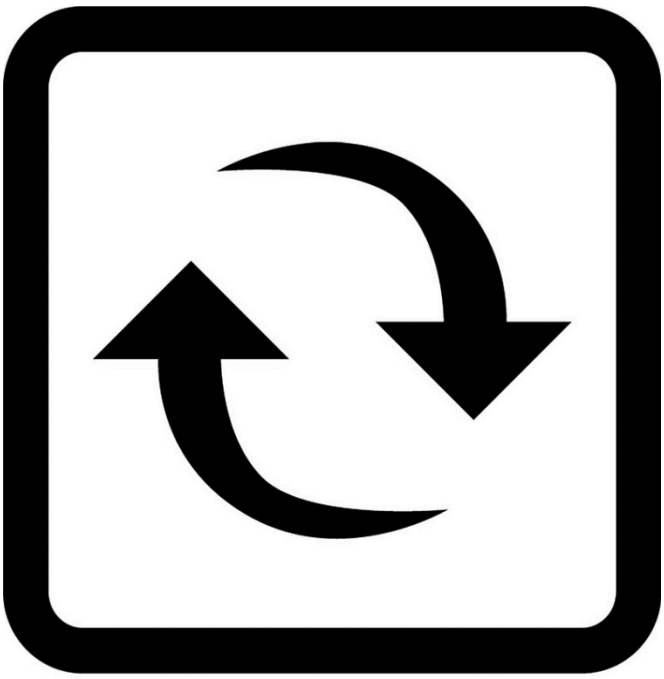

SERVERSCHECK

www.serverscheck.com

Quick Installation Guide

For Sensors with Nagios Core

Copyright:
Copyright © 2013 ServersCheck BVBA
All rights reserved.
Reproduction without permission is prohibited.

Software:
The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Trademarks:
ServersCheck is a trademark of ServersCheck.
All other trademarks or registered marks in this manual belong to their respective manufacturers.

Disclaimer:
Information in this document is subject to change without notice and does not represent a commitment on the part of ServersCheck.

ServersCheck provides this document “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. ServersCheck reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

ServersCheck has made this document to the best of its abilities. However ServersCheck assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

Document Overview

This document outlines the integration with the ServersCheck PoE & SNMP sensors with the open source and free Nagios Core software.

This document assumes that a sensor was configured on the network as per user manual instructions. The manual can be downloaded from www.serverscheck.com/sensors/manual.asp

This document also assumes that Nagios Core has been installed on a system. For more information on Nagios Core, please visit http://www.nagios.org/projects/nagioscore

SensorGateway SNMP Configuration

For uniformity and a simplified management, the following default parameters would be consistently used:

SNMP Port: 161

SNMPv2 SensorGateway: 192.168.11.20
SNMPv2 Read Community: Public

SNMPv3 SensorGateway: 192.168.11.30
SNMPv3 Username: Serverscheck
SNMPv3 Auth Key: auth12345
SNMPv3 Auth Protocol: MD5
SNMPv3 Privacy Key: priv12345
SNMPv3 Protocol: AES

Section 1 – Management Information Base of ServersCheck SensorGateway

Throughout this guide, for each SNMP command, we'll be using 2 different methods to specify an object ID - through numerics and the ServersCheck MIB as defined in your Linux system.

In this section, we'll learn how to utilize the ServersCheck MIB for your SensorGateway to make its OID's more readable. If one opted to settle for the numerical tradition, you can simply proceed to section 2.

ServersCheck MIB

Firstly, go ahead and download the MIB named SERVERSCHECK-MIB.txt **here:**
Insert Link Here

Having done that, place the MIB in the /usr/share/snmp/mibs **directory. Now despite the file being transferred there, we would still have to register it to the Net-SNMP index file for it to recognize our new MIB. There are two ways to do this:**

- **By doing an SNMP command to the SensorGateway, we effectively trigger the index file to recognize its associated MIB, causing it to be registered.**
- **By statically editing the index file**

For the first method, we won't be able to use an SNMP Get since we don't have our OID's yet. Instead, we will be running an SNMP Walk by executing the following command:

V2: *snmpwalk 192.168.11.20 .1.3 -v 2c -c public*
V3: *snmpwalk 192.168.11.30 .1.3 -v 3 -l authPriv -u Serverscheck -a MD5 -A auth12345 -x AES -X priv12345*

After letting that process finish, we should verify the MIB index found here:

/var/lib/net-snmp/mib_indexes/

Inside the directory, you will find a *0* file. Open it with your favorite editor and look for the following entry:

SERVERSCHECK-MIB SERVERSCHECK-MIB.txt

If what's mentioned is present, we should already be able to use our newly registered MIB. And apparently as a second method, you can just statically include this entry in the index file. In the following section, we would be getting our sensor OID's by doing an SNMP Walk using the ServersCheck MIB.

Section 2 – Getting the OID

Now that we can use the ServersCheck MIB, we will now get the objects we want to monitor by grabbing their OID's.

Named OID

For the first part, we will utilize our MIB for a more organized experience. Now to begin the walking:

snmpwalk 192.168.11.20 .1.3 -v 2c -c public -m SERVERSCHECK-MIB

With the plethora of objects to choose from, let's pick out the highlighted objects as shown:

```
SERVERSCHECK-MIB::sensor1name.0 = STRING: "Int. Temp"
SERVERSCHECK-MIB::sensor1Value.0 = STRING: "21.93"
SERVERSCHECK-MIB::sensor1LastErrMsg.0 = STRING: "DOWN"
SERVERSCHECK-MIB::sensor1LastErrTime.0 = STRING: "29 November 2013,20:17:13"
SERVERSCHECK-MIB::sensor2name.0 = STRING: "Ext. Temp"
SERVERSCHECK-MIB::sensor2Value.0 = STRING: "20.50"
SERVERSCHECK-MIB::sensor2LastErrMsg.0 = STRING: "DOWN"
SERVERSCHECK-MIB::sensor2LastErrTime.0 = STRING: "04 December 2013,16:04:14"
SERVERSCHECK-MIB::sensor3name.0 = STRING: "Humidity"
SERVERSCHECK-MIB::sensor3Value.0 = STRING: "61.76"
```

Figure 2-1. Named OID – SNMPv2

The .1.3 on the syntax tells the SNMP Walk command where to begin. Apart from that, you'll only be shown very limited output. The same applies for the numbered OID.

```
[scadmin@Linux1 nagios]$ snmpwalk 192.168.11.20 -v 2c -c public -m SERVERSCHECK-MIB
RFC1213-MIB::sysDescr.0 = STRING: "Temperature & Sensor Gateway"
RFC1213-MIB::sysObjectID.0 = OID: SERVERSCHECK-MIB::serverscheck
RFC1213-MIB::sysUpTime.0 = Timeticks: (6764309) 18:47:23.09
RFC1213-MIB::sysContact.0 = STRING: "http://www.serverscheck.com"
RFC1213-MIB::sysName.0 = STRING: "SGW20"
RFC1213-MIB::sysLocation.0 = STRING: "Rack3_Point2"
[scadmin@Linux1 nagios]$
[scadmin@Linux1 nagios]$
[scadmin@Linux1 nagios]$
```

Figure 2-2. SNMPWalk Limited Output

By using the ServersCheck MIB, we will be able to easily look for the sensors we want to monitor. For SNMPv3, the syntax would be:

snmpwalk 192.168.11.30 .1.3 -v 3 -l authPriv -u Serverscheck -a MD5 -A auth12345 -x AES -X priv12345 -m SERVERSCHECK-MIB

For our SNMPv2 configuration throughout this guide, we'll be using the values of "Ext. Temp" and "Humidity" from Figure 2-1.

Notice that the only difference with these commands so far from our initial SNMP Walk is the addition of the MIB argument *–m SERVERSCHECK-MIB*

Numbered OID

In the next commands, to use the numbered naming system, we'll just be changing the MIB argument from SERVERSCHECK-MIB **to** RFC-1215

```
iso.3.6.1.4.1.17095.3.1.0 = STRING: "Int. Temp"
iso.3.6.1.4.1.17095.3.2.0 = STRING: "19.92"
iso.3.6.1.4.1.17095.3.3.0 = STRING: "WARNING"
iso.3.6.1.4.1.17095.3.4.0 = STRING: "04 December 2013,15:53:08"
iso.3.6.1.4.1.17095.3.5.0 = STRING: "Ext. Temp"
iso.3.6.1.4.1.17095.3.6.0 = STRING: "19.75"
iso.3.6.1.4.1.17095.3.7.0 = STRING: "DOWN"
iso.3.6.1.4.1.17095.3.8.0 = STRING: "04 December 2013,15:53:09"
```

Figure 2-3. Numbered OID – SNMPv3

In our usage of this format, keep in mind that you can simply replace iso **with a** .1 **value. In that case, the OID for the external temperature's value would also be .1.3.6.1.4.1.17095.3.6.0**

For the SNMPv3 configuration, we'll use the "Int. Temp" and "Ext. Temp" values throughout the guide.

Now that we have the OID's of the sensors we want to monitor, we can now move on to make our Nagios configuration files.

Section 3 – SNMPv2 Configuration

More than the packaged plugins from Nagios, we can utilize external commands to be used for checking. In this guide, we'll be using the two most popular and common SNMP pollers on Nagios: the check_snmp Nagios plugin and *snmpget* command-line tool from Net-SNMP (usually already preinstalled on modern and updated Linux distros).

As a starter, we'll be utilizing the template linux-server host definition, which in turn uses the generic-host definition as found on the templates.cfg file in /etc/nagios/objects

To start with, let's define the host just like what we would do with any other Nagios checks.

```
define host {
use                     linux-server
host_name               SGW20-SNMPv2
alias                   SGW20-SNMPv2
address                 192.168.11.20
}
```

Service Configuration for SNMPv2

Before using the SNMP plugin, it is best advised to check your syntax first by directly executing it on the terminal. For the external temperature, we could test it by first going to your Nagios plugins directory and issuing this command:

./check_snmp -H 192.168.11.20 -o SERVERSCHECK-MIB::sensor2Value.0 -C public

With that, we should get the following output:

```
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.20 -o SERVERSCHECK-MIB::sensor2Value.0 -C public
SNMP OK - "21.06" |
```

Figure 3-1. check_snmp ServersCheck MIB – Ext. Temp value

Also, to be assured that this is indeed the external temperature sensor's value, we should verify by replacing the sensor2Value.0 field with sensor2name.0 and retrieve this information:

```
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.20 -o SERVERSCHECK-MIB::sensor2Value.0 -C public
SNMP OK - "21.06" |
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.20 -o SERVERSCHECK-MIB::sensor2name.0 -C public
SNMP OK - "Ext. Temp" |
```

Figure 3-2. check_snmp ServersCheck MIB – Ext. Temp name

For the numbered OID, the external temperature's name and value would be .1.3.6.1.4.1.17095.3.5.0 **and** .1.3.6.1.4.1.17095.3.6.0 **respectively as seen here:**

```
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.20 -o .1.3.6.1.4.1.17095.3.5.0 -C public
SNMP OK - "Ext. Temp" |
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.20 -o .1.3.6.1.4.1.17095.3.6.0 -C public
SNMP OK - "21.00" |
```

Figure 3-3. check_snmp Numbered OID – Ext. Temp

Notice how the codes are different from Net-SNMP's like the SNMP Walk we previously did. Also take note that in the following commands we'll be using – check_snmp **and** snmpget **– we won't need to specify an argument for the ServersCheck MIB anymore as that'll only be needed upon doing an SNMP Walk.**

For the Nagios plugins directory, on CentOS, RHEL, and Fedora, it usually is located in the /usr/lib/nagios/plugins/ **directory. If not yet installed, you can do so with this command**

sudo yum install nagios-plugins

Now that we have that, we can now define our external temperature sensor as our first service for the SensorGateway running SNMPv2 via the check_snmp **plugin. For our thresholds, we'll use 28 and 32 for warning and critical respectively. Notice that unlike in the command line tool, we won't need to specify our host on the syntax since that's already defined on the** host_name **field. The same difference would be noticed upon configuring SNMPv3.**

```
define service {
use                     generic-service
host_name               SGW20-SNMPv2
service_description     Temp-Ext:check_snmp
check_command           check_snmp!-o SERVERSCHECK-MIB::sensor2Value.0 -C public –w 28 –c 32
normal_check_interval   1
retry_check_interval    1
}
```

External Command Configuration for SNMPv2

Now let's add an external command that checks the humidity sensor's values via the *snmpget* command. To check for our OID's as shown previously, the following is done.

```
[scadmin@Linux1 nagios]$ snmpget 192.168.11.20 SERVERSCHECK-MIB::sensor3name.0 -v 2c -c public
SERVERSCHECK-MIB::sensor3name.0 = STRING: "Humidity"
[scadmin@Linux1 nagios]$ snmpget 192.168.11.20 SERVERSCHECK-MIB::sensor3Value.0 -v 2c -c public
SERVERSCHECK-MIB::sensor3Value.0 = STRING: "65.57"
[scadmin@Linux1 nagios]$ snmpget 192.168.11.20 .1.3.6.1.4.1.17095.3.9.0 -v 2c -c public
SNMPv2-SMI::enterprises.17095.3.9.0 = STRING: "Humidity"
[scadmin@Linux1 nagios]$ snmpget 192.168.11.20 .1.3.6.1.4.1.17095.3.10.0 -v 2c -c public
SNMPv2-SMI::enterprises.17095.3.10.0 = STRING: "65.57"
```

Figure 3-4. snmpget Humidity

For the configuration, we'll first define an external command on Nagios and then link or map it to another service configuration.

```
define command {
command_name    NET-SNMP-SGW20
command_line    snmpget 192.168.11.20 SERVERSCHECK-MIB::sensor3Value.0 -v 2c -c public
}
```

Having the external command defined, we would then have to associate it with another service.

```
define service {
use                     generic-service
host_name               SGW20-SNMPv2
service_description     Humidity:Net-SNMP
check_command           NET-SNMP-SGW20
normal_check_interval   1
retry_check_interval    1
}
```

Section 4 - SNMPv3 Configuration

If you want your monitoring solution secured with authentication and encryption, you can do so by opting for SNMPv3. For this setup, we will be utilizing the *AuthPriv* security level.

Firstly, on the SensorGateway, change the SNMP agent version to SNMPv3 and make sure to check its respective parameters as shown on the "SensorGateway SNMP Configuration" section on the beginning of this guide.

Then on Nagios, our configuration will be a little bit different due to some additional codes we have to add for our security parameters.

For this section, we'll be using a different SensorGateway with the following settings:

```
define host {
use                     linux-server
host_name               SGW30-SNMPv3
alias                   SGW30-SNMPv3
address                 192.168.11.30
}
```

Service Configuration for SNMPv3

Now for the first service, let's check the internal temperature value via the check_snmp **plugin. As advised previously, we'll begin by checking the OID on the terminal with this command:**

./check_snmp -H 192.168.11.30 -o SERVERSCHECK-MIB::sensor1Value.0 -P 3 -L authPriv -U Serverscheck -a MD5 -A auth12345 -x AES -X priv12345

```
[scadmin@Linux1 /]$ cd /usr/lib64/nagios/plugins
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.30 -o SERVERSCHECK-MIB::sensor1name.0 -P 3
-L authPriv -U Serverscheck -a MD5 -A auth12345 -x AES -X priv12345
SNMP OK - "Int. Temp" |
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.30 -o SERVERSCHECK-MIB::sensor1Value.0 -P 3
 -L authPriv -U Serverscheck -a MD5 -A auth12345 -x AES -X priv12345
SNMP OK - "21.80" |
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.30 -o .1.3.6.1.4.1.17095.3.1.0 -P 3
-L authPriv -U Serverscheck -a MD5 -A auth12345 -x AES -X priv12345
SNMP OK - "Int. Temp" |
[scadmin@Linux1 plugins]$ ./check_snmp -H 192.168.11.30 -o .1.3.6.1.4.1.17095.3.2.0 -P 3
-L authPriv -U Serverscheck -a MD5 -A auth12345 -x AES -X priv12345
SNMP OK - "21.86" |
```

Figure 4-1. check_snmp verification – SNMPv3

And for the actual configuration:

```
define service {
use                     generic-service
host_name               SGW30-SNMPv3
service_description     Temp-Int:AP-check_snmp
check_command           check_snmp!–o SERVERSCHECK-MIB::sensor1Value.0 -P 3
                        –L authPriv –U Serverscheck –a MD5 –A auth12345 –x AES
                        –X priv12345 –w 28 –c 32
normal_check_interval   1
retry_check_interval    1
}
```

Notes:

- **Make sure to double check your syntax especially on the** check_command **field since these are case-sensitive codes.**
- **Notice that in SNMPv2, we didn't have to specify it as version 2, unlike here where we are required to explicitly declare that we're using version 3.**
- **The whole entry for the** check_command **value would be the following:**

check_snmp!–o SERVERSCHECK-MIB::sensor1Value.0 -P 3 –L authPriv –U Serverscheck –a MD5 –A auth12345 –x AES –X priv12345 –w 28 –c 32

External Command Configuration for SNMPv3

For the external command, we'll be monitoring the external temperature sensor value with snmpget**. For our syntax:**

snmpget 192.168.11.30 SERVERSCHECK-MIB::sensor2Value.0 -v 3 -l authPriv -u Serverscheck -a MD5 -A auth12345 -x AES -X priv12345

```
[scadmin@Linux1 nagios]$ snmpget 192.168.11.30 SERVERSCHECK-MIB::sensor2name.0 -v 3 -l authPriv
-u Serverscheck -a md5 -A auth12345 -x aes -X priv12345
SERVERSCHECK-MIB::sensor2name.0 = STRING: "Ext. Temp"
[scadmin@Linux1 nagios]$ snmpget 192.168.11.30 SERVERSCHECK-MIB::sensor2Value.0 -v 3 -l authPriv
-u Serverscheck -a md5 -A auth12345 -x aes -X priv12345
SERVERSCHECK-MIB::sensor2Value.0 = STRING: "21.69"
[scadmin@Linux1 nagios]$
[scadmin@Linux1 nagios]$
[scadmin@Linux1 nagios]$ snmpget 192.168.11.30 .1.3.6.1.4.1.17095.3.5.0 -v 3 -l authPriv
-u Serverscheck -a md5 -A auth12345 -x aes -X priv12345
SNMPv2-SMI::enterprises.17095.3.5.0 = STRING: "Ext. Temp"
[scadmin@Linux1 nagios]$ snmpget 192.168.11.30 .1.3.6.1.4.1.17095.3.6.0 -v 3 -l authPriv
-u Serverscheck -a md5 -A auth12345 -x aes -X priv12345
SNMPv2-SMI::enterprises.17095.3.6.0 = STRING: "21.75"
```

Figure 4-2. snmpget verification – SNMPv3

Now that that's verified, we can now proceed to our configuration.

```
define command {
command_name        NET-SNMP-SGW30
command_line        snmpget 192.168.11.30 SERVERSCHECK-MIB::sensor2Value.0 –v 3
                    –l authPriv –u Serverscheck –a MD5 –A auth12345 –x AES
                    –X priv12345
}
```

Notes:

- **Notice how the codes from** snmpget **and** check_snmp **are a little bit different to each other.**
- **Also for the authentication and privacy protocols used, which is MD5 and AES respectively, the value isn't case sensitive. So for example, these could be inputted as md5, MD5, aes, or AES.**

Now to associate the defined command with another service:

```
define service {
use                         generic-service
host_name                   SGW30-SNMPv3
service_description         Temp-Ext:AP-Net-SNMP
check_command               NET-SNMP-SGW30
normal_check_interval       1
retry_check_interval        1
}
```

And there we have it. Basic configuration for monitoring the ServersCheck SensorGateway sensors are now good to go in your Nagios system.

To give an example, here's how it will look like:

SGW15_SensorHub-SNMPv3	Airflow	OK	12-10-2013 01:02:19	0d 2h 1m 19s	1/3	SNMP OK - "0.00"
	Fuel_Flow	OK	12-10-2013 01:02:35	0d 2h 1m 33s	1/3	SNMP OK - "-0.00"
	Fuel_Level	OK	12-10-2013 01:02:07	0d 2h 1m 31s	1/3	SNMP OK - "84.87"
	Motion_Sensor	OK	12-10-2013 01:02:35	0d 2h 1m 33s	1/3	SNMP OK - "OFF"
	PING	OK	12-10-2013 01:01:22	0d 2h 1m 16s	1/3	PING OK - Packet loss = 0%, RTA = 2.85 ms
	Power_Failure	OK	12-10-2013 01:02:35	0d 2h 1m 33s	1/3	SNMP OK - "PWR OK"
	Shock	OK	12-10-2013 01:02:19	0d 2h 1m 19s	1/3	SNMP OK - "1.01"
	Temp_Internal	OK	12-10-2013 01:02:19	0d 2h 1m 19s	1/3	SNMP OK - "24.40"
	Temperature_External	OK	12-10-2013 01:02:19	0d 2h 1m 19s	1/3	SNMP OK - "21.81"
	Water_Leak	OK	12-10-2013 01:02:35	0d 2h 1m 33s	1/3	SNMP OK - "DRY"
SGW20-SNMPv2	Humidity:Net-SNMP	OK	12-10-2013 01:02:35	0d 2h 1m 33s	1/3	SERVERSCHECK-MIB::sensor3Value.0 = STRING: "64.83"
	Temp-Ext:check_snmp	CRITICAL	12-10-2013 01:02:19	0d 0h 15m 49s	3/3	SNMP CRITICAL - *33.31*
SGW30-SNMPv3	PING	OK	12-10-2013 00:59:01	0d 5h 14m 26s	1/3	PING OK - Packet loss = 0%, RTA = 9.71 ms
	Temp-Ext:AP-Net-SNMP	OK	12-10-2013 01:02:35	0d 2h 1m 33s	1/3	SERVERSCHECK-MIB::sensor2Value.0 = STRING: "22.87"
	Temp-Int:AP-check_snmp	WARNING	12-10-2013 01:02:19	0d 0h 14m 19s	3/3	SNMP WARNING - *29.37*

The highlighted parts are the output of the initial configuration made and as we see here, the declared thresholds are working as expected.

For more information on other check_snmp **options, kindly refer here: https://www.nagios-plugins.org/doc/man/check_snmp.html**

For the Net-SNMP SNMPGet command-line tool, you can do *snmpget –help* **for more information.**

Available in our website is some predefined SensorGateway template configuration files for Nagios. To download, please go to the following link: